

Definiciones

Transcripción

[00:00] Les doy la bienvenida a alumnos y alumnas a nuestra aula número 2, en la cual estaremos hablando entonces sobre la manipulación de la base de datos. Trataremos algunas definiciones importantes para poder entonces proceder a trabajar con SQL. Entonces MySQL. ¿Cómo está organizada una base de datos?

[00:20] Es la primera pregunta que debemos responder o mejor dicho, lo que queremos hacer a través de esta clase es entender esta organización de la base de datos. En toda base de datos hay una entidad más grande, la principal de todas, que se llama base de datos, y esta entidad es un repositorio que almacena datos que pueden ser consultados.

[00:52] Y estos datos o esta base de datos entonces se almacena, por ejemplo en un disco duro, en una SSD. Es un espacio de memoria especial para guardar los datos. Entonces esta base de datos también está constituida por diversas entidades que se encargan de organizar los datos al interior de la base de datos y es sobre estas diversas entidades que estaremos hablando.

[01:23] La entidad más importante de la base de datos es la tabla. Cuando hablamos de una tabla debemos pensar, por ejemplo, en una planilla de Excel. De repente ustedes ya observan la planilla Excel que tiene columnas y que tiene filas pero cuando estamos hablando de una tabla en MySQL es un objeto que estaremos creando en el cual definiremos previamente o predefiniremos mejor el número de columnas que va a tener y el tipo de cada columna.



[01:57] Y también, a medida que se ingresen nuevas filas a nuestra columna, entonces también nuestra tabla va a crecer. Entonces veamos con detenimiento la tabla. La tabla está compuesta por los siguientes elementos. Las columnas, que para MySQL se conocen como campos, y las filas que en MySQL o en cualquier lenguaje SQL son los registros.

[02:30] Entonces aquí aparecen campos y registros. Vamos a hablar, cuando hablemos de campos, ustedes ya van a entender que estamos haciendo alusión a las columnas, y cuando hablemos de registros, entonces ustedes ya van a saber que estamos hablando de las filas. Entonces campos y filas.

[02:50] Estos principios se aplican a cualquier modelo SQL, ahí ya depende del tipo de software o el tipo de IDE que ustedes quieran usar. Puede ser Oracle, puede ser SQL Server, puede ser también Postgre SQL, cualquier tipo de SQL va a manejar el mismo padrón.

[03:16] Entonces, tenemos nuestros campos y nuestros registros, y al momento de crear nuestra tabla es mandatorio, es necesario que nosotros definamos primeramente cuántos campos vamos a tener, la cantidad de campos que tendremos y el tipo de datos que estarán depositados en cada campo.

[03:38] Yo no puedo, por ejemplo, ponerme a decir que este campo va a ser un campo numérico, supongamos, el campo A es un campo numérico y eventualmente en el campo A yo digité letras o caracteres de texto, porque mi base de datos va a generar un error. Entonces nosotros siempre debemos respetar el parámetro que fue establecido para el tipo de datos que se van a consignar en ese campo determinado.

[04:08] Hay diversos tipos de campo, no vamos a entrar en detalle en este momento, pero si ustedes pueden tener una idea que podemos tener campos de texto que pueden contener variables de tipo char, del tipo varchar, del tipo text, después hablaremos más detalladamente sobre ellas.



[04:26] También tenemos campos numéricos que pueden tener variables de tipo int, son íntegros, bigint, números íntegros grandes o números integrales grandes, smallint, float, que tienen un punto float, generalmente son casillas decimales. Y bool lo puse numérico, a pesar de ser un tipo de campo lógico porque cuando estamos hablando de booleanos tiene 1 o 0.

[04:54] 1 cuando es verdadero, cuando una condición es verdadera, o 0 cuando es falso. Bien, también podemos tener un campo de fecha, que puede ser a través de variables tipo date, tipo datetime, timestamp, etcétera. Como les mencioné anteriormente, la información debe introducirse de acuerdo con la definición del campo, esto es importantísimo.

[05:18] Digamos si yo defino un campo que va a ser bool, yo no puedo colocar un número 2. Únicamente ese campo va a admitir 0 o 1. Entonces esa es la idea de forma global. Ahora bien, vamos a hablar sobre los campos. Los campos los definimos al momento de la creación de la tabla, esto es principal.

[05:42] Cuando creamos una tabla ya tenemos que definir nuestros campos y el tipo de campo que vamos a tener. Estos campos se limitan a lo que fue definido en su definición. Digamos yo, al momento de tener mi tabla, yo ya no puedo, por ejemplo, decir: "Me voy a voy a colocar estos registros, por ejemplo, en un campo inexistente". Eso no tiene sentido.

[06:05] Entonces mis campos tienen ese número limitado a lo que yo defina. Y no admite datos que no fueron especificados. Redundamos en este tema porque a veces la persona dice: "no, pero es que estoy tratando de digitar este registro y no lo admite, la base de datos me da error". Entonces es importante que sepamos todos, que el campo siempre va a exigir que se coloquen el tipo de datos que fue especificado.

[06:39] Ahora vamos a hablar sobre los registros. Los registros, ya, como les dije, son las filas, o sea, son los datos que se encuentran contenidos en cada campo. Es un número ilimitado de registros. Una tabla puede tener un número



ilimitado de registros, o sea, n cantidad de registros. Siempre será limitado lógicamente, por el espacio de almacenaje disponible en disco o también por una política de crecimiento de la base de datos.

[07:11] Esto es lo que limita realmente la cantidad de registros que yo puedo llegar a tener, pero si no fueran estas condiciones, la cantidad de registros podría ser ilimitada. Ahora que tenemos nuestros campos y registros, hay una parte muy importante de la tabla que es nuestro índice.

[07:32] El índice es simplemente un número que nos ayuda a localizar con facilidad un registro o una serie de registros al interior de mi tabla. Entonces, el índice tiene un algoritmo que lo que hace es localizar rápidamente algún elemento de mi tabla. Si no, ella tendría que avanzar de forma secuencial, ir encontrando cada registro, trayéndolo.

[08:01] No. Digamos, el índice ya me facilita esta búsqueda y lo hace más ágil. Bien. Ya que conocemos la composición de nuestra tabla, ahora vamos a hablar de elementos relevantes también en nuestra tabla. Un elemento importantísimo que es opcional, pero a la misma vez debemos tenerlo en cuenta son las claves primarias o la clave primaria.

[08:30] La clave primaria es un campo en el cual ninguno de sus registros se puede repetir en sí mismo. Es decir, si estoy hablando de una clave primaria, por ejemplo, el número de cédula o el número de identificación de un cliente, yo no puedo tener dos registros en mi lista de clientes con el mismo número de cédula.

[08:58] Entonces, mi clave primaria no me permite que este registro sea repetido. Ello con el objetivo de que haya integridad en mis datos, esa es la idea de una base de datos estructurada. Esa estructura nos ayuda a que nuestros datos estén completos, no vayamos a tener ninguna incoherencia o algún problema futuro con ellos.



[09:35] Mi clave primaria también, como les mencioné es opcional, no es obligatoria, pero hay muchos casos en los cuales sí necesitamos establecer una clave primaria. Para que no tengamos ese problema, el caso de registro de clientes, por ejemplo, es obligatorio porque, ¿cómo es que vamos a registrar a un mismo cliente dos veces?

[09:56] Entonces la clave primaria y también yo puedo tener una clave primaria compuesta por dos campos. Entonces la combinación de estos campos no se puede repetir en ningún otro registro de mi tabla. Ejemplo. Tengo el número de cédula y el registro de identificación tributaria o el número de identificación tributaria de un cliente.

[10:21] Entonces yo tengo el número de cédula de ese cliente y su identificación tributaria y la combinación de estos dos no se puede repetir en ninguna otra línea o en ninguna otra fila o en ningún otro registro de mi tabla. Entonces es importante saber esto. Ahora, también recordemos que una base de datos está compuesta por muchas tablas.

[10:44] Y en muchas tablas yo puedo tener una o más claves primarias, que también no son obligatorias, puedo tenerlas como no puedo tenerlas, pero para yo poder entonces relacionar cada una de mis tablas, recordemos que estamos hablando de bases de datos relacionales porque existe una relación entre ellas, entonces necesito una clave externa o una clave foránea, que es un campo que me va a permitir relacionar cada una de mis tablas entre sí.

[11:19] Entonces yo puedo hacer una conexión, crear un vínculo entre mis tablas a través de una clave externa. Entonces, es el campo que relaciona a mis tablas, estamos hablando de bases de datos relacionales. Antiguamente las bases de datos eran secuenciales y no existía este tipo de vínculo o tipo de relación entre los datos, y de forma secuencial, tenía que recorrer toda la base de datos hasta obtener el registro que se necesitaba. Ahora no.



[11:47] A través de este principio de bases de datos relacionales, entonces ya simplemente al tener una clave externa yo puedo relacionar una tabla con otras tablas. Entonces, ello también garantiza la integridad de datos, por ejemplo, si tengo mi cliente en una tabla y tengo mis productos en otra tabla, entonces, por ejemplo mi llave o mi clave externa puede ser el número de identificación tributaria tanto en mi tabla de producto como en mi tabla de clientes.

[12:23] Entonces yo no puedo venderle un producto a un número de identificación tributaria que no se encuentre en mi registro de clientes. Entonces, primero ese cliente debe estar registrado para yo poder realizarle una venta de un producto. Entonces, la clave externa es muy importante y ella nos permite que nuestros datos estén bien organizados.

[12:47] Que no haya duplicidad de la información, que todo esté en orden al interior de mi base de datos. Ahora vamos a hablar un poco sobre otras entidades importantes. Y por ejemplo, bueno, también, como les dije, una base de datos tiene muchas tablas, pero estas tablas yo las puedo también agrupar a través de esquemas.

[13:17] Entonces los esquemas son asuntos o temas que me facilitan a mí la organización al interior de mi base de datos. Entonces yo agrupo mis tablas de acuerdo con su tema. Esto no quiere decir que yo no pueda tener un esquema con todas las tablas de mi base de datos o cinco esquemas, eso no interesa. Eso es simplemente una forma de agrupación de tablas, pero puede alterarse de acuerdo con la necesidad.

[13:50] Ahora, también mi base de datos puede generar algo que se llama view, una entidad que se llama view o que se llama vista. Y esta vista se realiza de la siguiente forma: se consulta un número determinado de tablas al mismo tiempo y se trae la información.



[14:12] El problema con los views o con las vistas es que dependiendo de la función o de los comandos SQL que yo emplee para realizar este tipo de vista puede tener un costo muy alto de procesamiento, entonces puede que en algún punto no sea lo más recomendable, pero generalmente las views facilitan mucho tomar elementos de varias tablas.

[14:43] Porque yo no quiero simplemente tomar las tablas y unir las todas, porque de nada me sirve a mí tener un view con 10.000 campos. Yo simplemente quiero por ejemplo, tomar un view con la identificación del número de cédula de mi cliente, con los productos que compró, con la fecha que compró y si los tengo en diferentes tablas, entonces yo puedo crear un view haciendo estas consultas de la siguiente manera.

[15:07] Entonces tengo todas mis tablas. Entre mis tablas hay una relación a través de una clave externa, que me permite entonces realizar una consulta, la consulta o query, recuerden que estamos hablando de SQL: Structured Query Language, lenguaje de consulta estructurado.

[15:27] Entonces yo hago mi consulta de acuerdo con las relaciones que tengo entre mis tablas aquí y obtengo mi información. Esta consulta entre tablas la puedo hacer a través de un join, entonces yo junto mis tablas a través de una clave externa, que serían dos campos que coinciden en dos tablas diferentes.

[15:53] Y puedo al mismo tiempo así, ir filtrando también mi información, por ejemplo yo quiero obtener los clientes hombres que compraron un determinado producto o únicamente las mujeres o únicamente cierto rango de edades. Ahí ya creamos los filtros de acuerdo con nuestra necesidad y en ese momento entonces podemos tener toda la información en un solo lugar, la información que queremos en un instante a través de una vista.

[16:24] Ahora bien, nuestra base de datos también puede tener procedures o procedimientos. Estos procedimientos son lo mismo que una lógica estructurada que a través de un lenguaje nativo creado en MySQL. SQL como



tal, de acuerdo con el estándar ANSI no tenía ningún tipo de lógica, un lenguaje lógico de programación como varias opciones, sino que era únicamente para consultas.

[17:00] Pero los constructores de MySQL, de Oracle, de Postgre, todos ellos tuvieron en cuenta crear un lenguaje sencillo, basado en QSL, en comandos SQL, para poder hacer consultas o para mejor dicho colocar condiciones, colocar procedimientos de lazo, y así generar diferentes resultados. Entonces esa es la idea a través de los procedimientos y lógicamente, nuestros procedimientos nos generan funciones.

[17:35] Hay funciones ya propias de SQL que nos permiten, por ejemplo, establecer rangos de fechas, nos permiten quitar espacios, hay funciones que nos permiten también eliminar.

[17:52] Por ejemplo, hay casos en los cuales se presenta la necesidad de cambiar minúsculas a mayúsculas, o colocar todo en mayúscula, en fin, diversas funciones que ya vienen en SQL, pero también podemos crear funciones de acuerdo con nuestra necesidad, con una serie de comandos propios de nuestra base de datos.

[18:15] Ahora bien, también nuestra base de datos puede tener otra entidad llamada trigger o su traducción podría ser disparador. Lo que hace es como activar una alerta que me permita ejecutar algún tipo de cambio cuando se presente alguna modificación al interior de la base de datos.

[18:43] Entonces esta alerta lo que hace es crear, digamos en general puede producir una función. Digamos si yo tengo un trigger en mi base de datos de producto que se vendió un producto en ese momento, automáticamente este trigger se va a activar y en la base de datos de clientes le va a asignar al campo, donde dice el número de productos que ha comprado este cliente si esta persona compró.

[19:09] Entonces ahí se va a incrementar, por ejemplo, la cantidad de productos que compró. Fíjense la idea, que se active ese cambio. Entonces, ejecuta una función o procedimiento cuando la condición del trigger es satisfecha. ¿Compró productos aquí en la tabla de productos? Entonces ya mi tabla de clientes automáticamente se altera el campo de número de productos comprados.

[19:31] Más o menos esa es la idea con trigger. Ahora bien, vamos entonces a resumir aquí rápidamente, las entidades que componen nuestra base de datos. Tenemos nuestras tablas, muchas tablas. Con estas tablas podemos generar consultas y producir views.

[19:53] Tenemos nuestros triggers y tenemos también nuestros procedures. Entonces esto es de forma muy global, de forma bastante general para una